

Илья Дуванов

Параметры логирования

Обновлено 22 мая 2025, 16:19

Содержание страницы

- [1. API Gateway](#)
- [2. Service Prepare](#)
- [3. Kafka](#)
- [4. Service Predict](#)
- [5. Service Update Rentals](#)
- [6. Service Update Models](#)
- [7. PostgreSQL](#)
- [8. Redis](#)
- [Общая информация](#)

1. API Gateway

Цель логирования: Мониторинг маршрутизации запросов, производительности и распределения нагрузки между микросервисами.

Параметр	Описание	Тип данных	Пример значения	Назначение
timestamp	Время обработки запроса в API Gateway.	String	"2025-05-22T15:43:01Z"	Синхронизация событий для анализа задержек в цепочке обработки.
endpoint	Маршрут запроса (например, "/predict", "/update-actual").	String	"/predict"	Определение, какие маршруты наиболее нагружены (для оптимизации).
processing_time_ms	Время обработки запроса в API Gateway (в миллисекундах).	Integer	15	Оценка производительности API Gateway.
status_code	HTTP-код ответа (200 — успех, 404 — не найдено, 500 — ошибка сервера).	Integer	200	Диагностика ошибок (например, высокий % ошибок 500 может указывать на сбой).
subscription	Тип подписки пользователя (free/premium).	String	"premium"	Анализ распределения запросов по типам подписки (для бизнес-аналитики).

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:00Z",
  "client_ip": "192.168.1.1",
  "request_type": "predict",
  "user_id": "user123",
  "status": 200
}
```

Мониторинг:

- Дашборд Grafana: "Распределение запросов по endpoint'ам" (гистограмма по endpoint).
- Среднее время обработки (processing_time_ms) для выявления узких мест.
- Процент запросов по подписке (subscription) для оценки популярности платного режима.

2. Service Prepare

Цель логирования: Отслеживание процесса предобработки данных (нормализация, кодирование), отправляемых в Kafka для прогноза.

Параметр	Описание	Тип данных	Пример значения	Назначение

timestamp	Время выполнения предобработки.	String	"2025-05-22T15:43:02Z"	Синхронизация событий в цепочке обработки.
user_id	Уникальный идентификатор пользователя.	String	"user123"	Отслеживание, для кого выполнялась предобработка (для анализа ошибок).
subscription	Тип подписки (free/premium).	String	"premium"	Определение, для какого режима обрабатываются данные (free - общая модель).
input_features	Входные данные после предобработки (например, час, температура).	Object	{"hour": 12, "temperature": 20}	Проверка корректности предобработки (например, нет ли пропущенных данных).
processing_time_ms	Время предобработки (в миллисекундах).	Integer	10	Оценка производительности предобработки.

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:02Z",
  "user_id": "user123",
  "subscription": "premium",
  "input_features": {"hour": 12, "temperature": 20},
  "processing_time_ms": 10
}
```

Мониторинг:

- Проверка корректности входных данных через input_features (для отладки).
- Анализ времени обработки (processing_time_ms) для оптимизации предобработки.
- Распределение запросов по subscription (дашборд: "Free vs Premium запросы").

3. Kafka

Цель логирования: Мониторинг передачи сообщений между микросервисами через топики Kafka.

Параметр	Описание	Тип данных	Пример значения	Назначение
timestamp	Время отправки/ получения сообщения.	String	"2025-05-22T15:43:03Z"	Синхронизация событий в цепочке обработки.
topic	Название топика	String	"preprocessed-data-premium"	Отслеживание, в какой топик отправлено сообщение (для анализа нагрузки).
message_size_bytes	Размер сообщения в байтах.	Integer	256	Оценка объема данных, передаваемых через Kafka (для оптимизации).
status	Статус отправки/ получения (sent, failed).	String	"sent"	Диагностика проблем с передачей сообщений (например, высокий % failed).

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:03Z",
  "topic": "preprocessed-data-premium",
  "message_size_bytes": 256,
  "status": "sent"
}
```

Мониторинг:

- Распределение сообщений по топикам (topic) для анализа нагрузки (дашборд: "Сообщения по топикам").
- Средний размер сообщений (message_size_bytes) для оптимизации передачи данных.
- Процент ошибок (status: failed) для диагностики проблем с Kafka.

4. Service Predict

Цель логирования: Отслеживание выполнения прогнозов, использования кэша и производительности.

Параметр	Описание	Тип данных	Пример значения	Назначение
timestamp	Время выполнения прогноза.	String		Синхронизация событий.

user_id	Уникальный идентификатор пользователя.	String		Идентификация пользователя для анализа прогнозов.
subscription	Тип подписки (free/premium).	String		Определение, какая модель использовалась (общая или индивидуальная).
prediction	Прогнозируемое количество аренд (целевое значение).	Integer		Хранение результата прогноза для анализа точности (если есть фактические данные).
cache_source	Источник модели (memory – In-Memory Cache, redis – Redis, db – PostgreSQL).	String		Оценка эффективности кэширования.
processing_time_ms	Время выполнения прогноза (в миллисекундах).	Integer		Оценка производительности прогноза.

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:04Z",
  "user_id": "user123",
  "subscription": "premium",
  "prediction": 150,
  "cache_source": "redis",
  "processing_time_ms": 5
}
```

Мониторинг:

- Эффективность кэширования через cache_source (ashboard: "Попадания в кэш: memory/redis/db").
- Среднее время прогноза (processing_time_ms) для оптимизации.
- Сравнение прогнозов по subscription (free vs premium) для оценки точности.

5. Service Update Rentals

Цель логирования: Мониторинг обновления фактических данных в PostgreSQL.

Параметр	Описание	Тип данных	Пример значения	Назначение
timestamp	Время выполнения обновления.	String		Синхронизация событий.
request_id	Уникальный идентификатор запроса, для которого обновляются данные.	String	"12345"	Связь с прогнозом для анализа точности (прогноз vs фактическое значение).
actual_rentals	Фактическое количество аренд, предоставленное пользователем.	Integer	148	Хранение фактических данных для дообучения и анализа точности.
processing_time_ms	Время обновления записи (в миллисекундах).	Integer	8	Оценка производительности обновления.
status	Статус выполнения (success, failed).	String	"success"	Диагностика ошибок (например, "failed" при отсутствии request_id).

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:05Z",
  "request_id": "12345",
  "actual_rentals": 148,
  "processing_time_ms": 8,
  "status": "success"
}
```

Мониторинг:

- Количество обновлений фактических данных (ashboard: "Обновления в день").
- Процент ошибок (status: failed) для диагностики.
- Среднее время обновления (processing_time_ms).

6. Service Update Models

Цель логирования: Отслеживание процесса дообучения моделей (индивидуальных и, возможно, общей).

Параметр	Описание	Тип данных	Пример значения	Назначение
timestamp	Время выполнения дообучения.	String	"2025-05-22T15:43:06Z"	Синхронизация событий.
user_id	Идентификатор пользователя (или "general" для общей модели).	String	"user123"	Идентификация, для кого дообучается модель (или общая модель).
samples_used	Количество записей, использованных для дообучения.	Integer	100	Оценка объема данных для дообучения (например, достаточно ли данных).
processing_time_ms	Время дообучения модели (в миллисекундах).	Integer	200	Оценка производительности дообучения.
status	Статус выполнения (success, failed).	String	"success"	Диагностика ошибок (например, "failed" при недостатке данных).

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:06Z",
  "user_id": "user123",
  "samples_used": 100,
  "processing_time_ms": 200,
  "status": "success"
}
```

Мониторинг:

- Количество дообученных моделей (ashboard: "Дообучения в неделю").
- Среднее время дообучения (processing_time_ms) для оптимизации.
- Проверка качества данных через samples_used (например, если <50, дообучение неэффективно).

7. PostgreSQL

Цель логирования: Мониторинг операций с базой данных (чтение/запись моделей, прогнозов, фактических данных).

Параметр	Описание	Тип данных	Пример значения	Назначение
timestamp	Время выполнения операции.	String	"2025-05-22T15:43:07Z"	Синхронизация событий.
operation	Тип операции (insert, update, select).	String	"update"	Отслеживание типов запросов для анализа нагрузки на базу.
table	Название таблицы (например, predictions, models).	String	"predictions"	Определение, какая таблица используется (для оптимизации структуры).
rows_affected	Количество затронутых строк.	Integer	1	Оценка объема операций (например, массовое обновление или одиночное).
execution_time_ms	Время выполнения запроса (в миллисекундах).	Integer	5	Оценка производительности базы

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:07Z",
  "operation": "update",
  "table": "predictions",
  "rows_affected": 1,
  "execution_time_ms": 5
}
```

Мониторинг:

- Распределение операций по типам (operation) и таблицам (table).
- Среднее время выполнения (execution_time_ms) для оптимизации запросов.

- Количество затронутых строк (rows_affected) для анализа нагрузки.

8. Redis

Цель логирования: Мониторинг использования кэша для индивидуальных моделей.

Параметр	Описание	Тип данных	Пример значения	Назначение
timestamp	Время выполнения операции с кэшем.	String	"2025-05-22T15:43:08Z"	Синхронизация событий.
operation	Тип операции (get, set, delete).	String	"get"	Отслеживание операций с кэшем (например, частота очистки).
key	Ключ в Redis (например, "model:user123").	String	"model:user123"	Идентификация, какая модель запрашивается/сохраняется.
cache_hit	Была ли модель найдена в кэше (true — найдена, false — загружена из базы).	Boolean	true	Оценка эффективности кэширования
processing_time_ms	Время выполнения операции (в миллисекундах).	Integer	1	Оценка производительности Redis

Пример лога:

```
{
  "timestamp": "2025-05-22T15:43:08Z",
  "operation": "get",
  "key": "model:user123",
  "cache_hit": true,
  "processing_time_ms": 1
}
```

Мониторинг:

- Процент попаданий в кэш (cache_hit) для оценки эффективности.
- Среднее время доступа (processing_time_ms) для сравнения с PostgreSQL.
- Частота операций (operation) для анализа нагрузки на Redis.

Общая информация

Все логи в формате JSON для удобства парсинга в Loki.

Ротация логов Loki - 7 дней